

# 工作

这个星期首先尝试了从严丙辉提供的网站上爬取地球纹理数据。利用自己编写的 HTTP 下载代码，爬到了 0~3 层纹理。后来考虑到如果全部爬取需要数 G 的硬盘空间，也没有必要，所以停止了。

星期二朱标将调试好的 MPI 代码交付了给我。由于并行框架设计不完善，有大量改进，所以代码合并费了两天时间。合并过程中做了几个工作：

- 1、标准化了 IResource 接口，使得资源可以在节点(Slave)之间，通过 MPI 分段传输（原来对于资源的定义非常凌乱，并且无法传输，传输也是整体传输，不能分段，没有考虑到 TCP/IP 每个包 512 字节的限制）。
- 2、标准化了 IMessengerMaster、IMessengerSlave 等接口，现在只要替换 Local 的实现和 MPI 的实现，就可以在本地单进程多线程并行和分布式多机并行之间切换。

最后剩下的几天进一步完善了框架，并且开始编写基于 Object Space 的并行光线投射体绘制程序。除了原有存在于 System Memory 中的体数据资源，现在开始需要支持显存中的体数据资源，以及 RenderTarget 资源。这些资源都将实现通过 MPI 的数据传输。然后做体绘制。

# 新的框架

test_parallel_mpi			
parallel_framework	parallel_resource	parallel_messenger_mpi	
solsys			
Boost		stl	MPICH2

test_parallel_graphics_draw_quad				
parallel_framework	parallel_resource	parallel_messenger_local	graphics_opengl	
solsys				
Boost		stl	OpenGL	

现在总共有四层。底层是 Boost、STL、MPICH2 和 OpenGL。Solsys 相当于 Boost、STL 的扩展，提供了额外的基本容器、数学计算等功能；同时还定义了许多接口，如网络接口、图形接口、数据库接口。再上一层是实现层，实现了 solsys 中的接口。顶层是应用层，各个应用程序基于实现层和 solsys，成为一个完整的应用程序。

上图中 parallel\_messenger\_mpi 和 parallel\_messenger\_local 是 solsys 的 parallel 中 messenger 的两种实现，分别用于 MPICH2 环境下的分布式并行，和单机单进程下的多线程并行。图中的两个应用程序只要简单替换 parallel\_messenger\_mpi 和 parallel\_messenger\_local 就可以在不同的环境下运行。

Parallel\_resource 中定义并且实现了所有的资源。如果有新的资源类型，全部添加到这个库中。

Parallel\_framework 是并行的框架，实现了调度(scheduler)、代价模型(用于资源分配和负载平衡)、内存管理策略(ResourceManager::TimeToRemove)。需要研发以上几个算法时，可以通过这里的代码完成。

# 接下来的工作

考虑到前几天的通知说硕士毕业论文截止日期是 1 月下旬，所以计划再花 1 星期时间将并行体绘制实现，然后一边写毕业论文，一边完善程序，做各种测试。

基于 Image Space 的并行地球绘制工作和地球纹理已经交给朱标，并提供了简单的面绘制程序做参考。